WEEK 2

21/12/23

Program 1: WAP that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.*;
public class Quad
{
  public static void main(String args[])
  {
    float a, b, c, d, r1, r2;
    System.out.println("Enter a, b, c : ");
    Quad obj = new Quad(); Scanner read = new Scanner(System.in);
    a = read.nextFloat();
    b = read.nextFloat();
    c = read.nextFloat();
    d = 0.0f;
    r1 = 0.0f;
    r2 = 0.0f;
    if (a==0 || b==0 || c==0)
    {
     System.out.println("Invalid input");
    }
    else
    {
      d = b*b - 4*a*c;
      if (d>0)
      {
        r1 = (float)(-b + Math.sqrt(d)) / (2*a);
        r2 = (float)(-b - Math.sqrt(d)) / (2*a);
        System.out.println("Roots are real and distinct \n
                            R1 = " + r1 + "\t R2 = " + r2);
      }
    else if (d<0)
      {
```

```
        System.out.println("Roots are imaginary");
    }
    else
    {
        r1 = -b/(2*a);
        r2 = r1;
        System.out.println("Roots are equal in R1 = " + r1 +
                           " & R2 = " + r2);
    }
}
}
```

Output :-

(iii) Enter values of a, b, c:

21

38

15

Roots are real and distinct

R1 = - 0.58179384 R2 = -1.22773

(iv) Enter values of a, b, c:

10

0

12

Invalid input

(v) Enter values of a, b, c:

2

1

2

Roots are imaginary

22/12

```
C:\Users\BMSCE\Desktop\1BMSSCS014>javac QuadEq.java

C:\Users\BMSCE\Desktop\1BMSSCS014>java QuadEq
Enter values of a, b, c:
2
1
2
Roots are imaginary
Name: Aditi C
USN: 1BM22CS014
```

```
C:\Users\BMSCE\Desktop\1BMSSCS014>javac QuadEq.java

C:\Users\BMSCE\Desktop\1BMSSCS014>java QuadEq
Enter values of a, b, c:
1
6
1
Roots are real and distinct
R1= -0.17157288 R2= -5.8284273
Name: Aditi C
USN: 1BM22CS014
```

```
C:\Users\BMSCE\Desktop\1BMSSCS014>javac QuadEq.java

C:\Users\BMSCE\Desktop\1BMSSCS014>java QuadEq
Enter values of a, b, c:
10
0
12
Invalid Input
Name: Aditi C
USN: 1BM22CS014
```

```
C:\Users\BMSCE\Desktop\1BMSSCS014>javac QuadEq.java

C:\Users\BMSCE\Desktop\1BMSSCS014>java QuadEq
Enter values of a, b, c:
1
2
1
Roots are real and equal
R1= -1.0         R2= -1.0
Name: Aditi C
USN: 1BM22CS014
```

```
C:\Users\BMSCE\Desktop\1BMSSCS014>javac QuadEq.java

C:\Users\BMSCE\Desktop\1BMSSCS014>java QuadEq
Enter values of a, b, c:
21
38
15
Roots are real and distinct
R1= -0.58179384 R2= -1.22773
Name: Aditi C
USN: 1BM22CS014
```

Program 2 : WAP to create a class Student with members USN, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a Student.

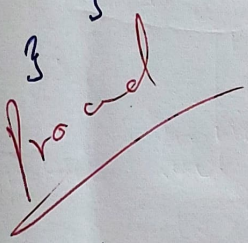
$$SGPA = \frac{\text{Total GP}}{\text{Total credits}}$$

```
import java.util.*;
class Student {
    String name, usn;
    int credit[], marks[];
    public Student (int credit[], int marks[], String name,
                    String usn)
    {
        this.usn = usn;
        this.name = name;
```

```
        this.credit = credit;
        this.marks = marks;
    }

    public void display (double res)
    {
        System.out.println("\n Name : " + name);
        System.out.println("\n USN : " + usn);
        System.out.println("\n Credits Marks : "); ("\n Marks & marks:");
        for(int i=0 ; i< marks.length; i++)
        {
            System.out.println("\n" + credit[i]);
            System.out.println("\n Sub " + (i+1) + "\t Marks = " + marks[i]
                               + "\t Credits = " + credit[i]);
        } System.out.println("\n\n SGPA = " + res);
    }

    public double sgpa()
    {
        double tc = 0 ; //total credits
        double tgp = 0 ; //total grade points

        for(int i=0; i< credit.length ; i++)
        {
            tc += credit[i];
            tgp += calgp(marks[i]) * credit[i];
        }
        return (tgp/tc);
    }

    public double calgp (int m)
    {
        if(m>=90)
```

```
    {
      return 10.0;
    }
  else if (m>=80)
    {
      return 9.0;
    }
  else if (m>=70)
    {
      return 8.0;
    }
  else if (m>=60)
    {
      return 7.0;
    }
  else if (m>=50)
    {
      return 6.0;
    }
  else
    {
      return 0.0;
    }
}

public static void main(String args[]) {
  Student ob = new Student();
  Scanner read = new Scanner (System.in);
  usn = read
  System.out.println("\n Enter name : ");
  name = read.nextLine();
  System.out.println("\n Enter usn : ");
  usn = read.next();
  System.out.println("\n Enter no. of subjects : ");
  int n = read.nextInt();
  System.out.println("
```

```
credit = new int[n];
marks = new int[n];
System.out.println("\n Enter marks and credits \n:");
for(int i=0; i<n; i++)
{
    System.out.println("\nMarks : ");
    credit[i] = read.nextInt();
    System.out.println("\n Credit : ");
    marks[i] = read.nextInt();
}
Student ob = new Student(credit, marks, name, usn);
double sres = ob.sgpa();
ob.display(sres);
}
}
```



Output :

```
Enter name : ABC
Enter USN : 001
Enter marks & credits :
Marks : 90
Credit : 4
Marks : 86
Credit : 3
Marks : 95
Credit : 4
Marks : 89
Credit : 1
Marks : 98
Credit : 1
```

```
Name : ABC
USN : 001
Ex
```

Algorithm :-

Step 1 : Start

Step 2 : input usn, name, ~~credits, marks~~, n

Step 3 : for i=0 to n-1 , do
read credits[i], marks[i]
endoffor

Step 4 : for j=0 to n-1 ; do
tc = tc+ credits[i];
tgp= tgp+(callfunction calgp(marks[j]) * credit[j]);
endoffor

Step 5 : res = tc / tgp

Step 6 : output usn, name

Step 7 : for k=0 to n-1, do
output credits[k], marks[k]
endoffor

Step 8 : output res

Step 9 : stop

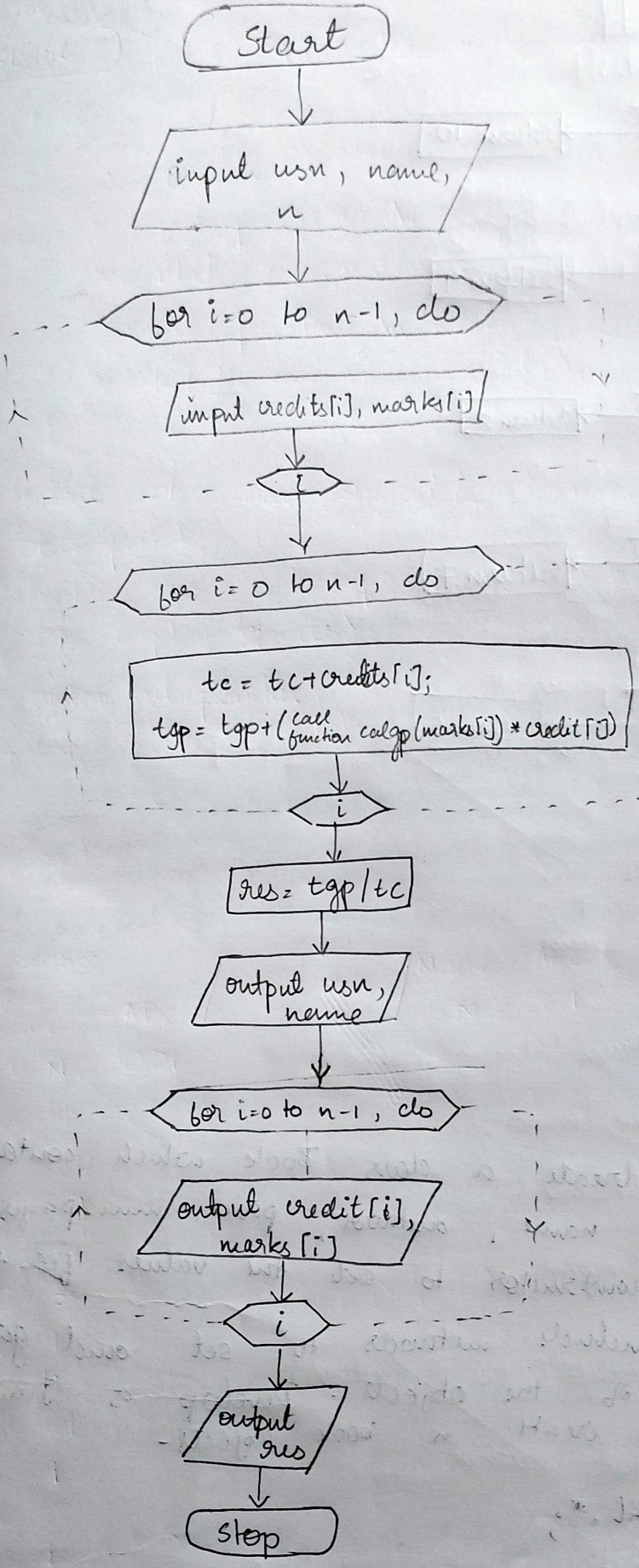
calgp function:

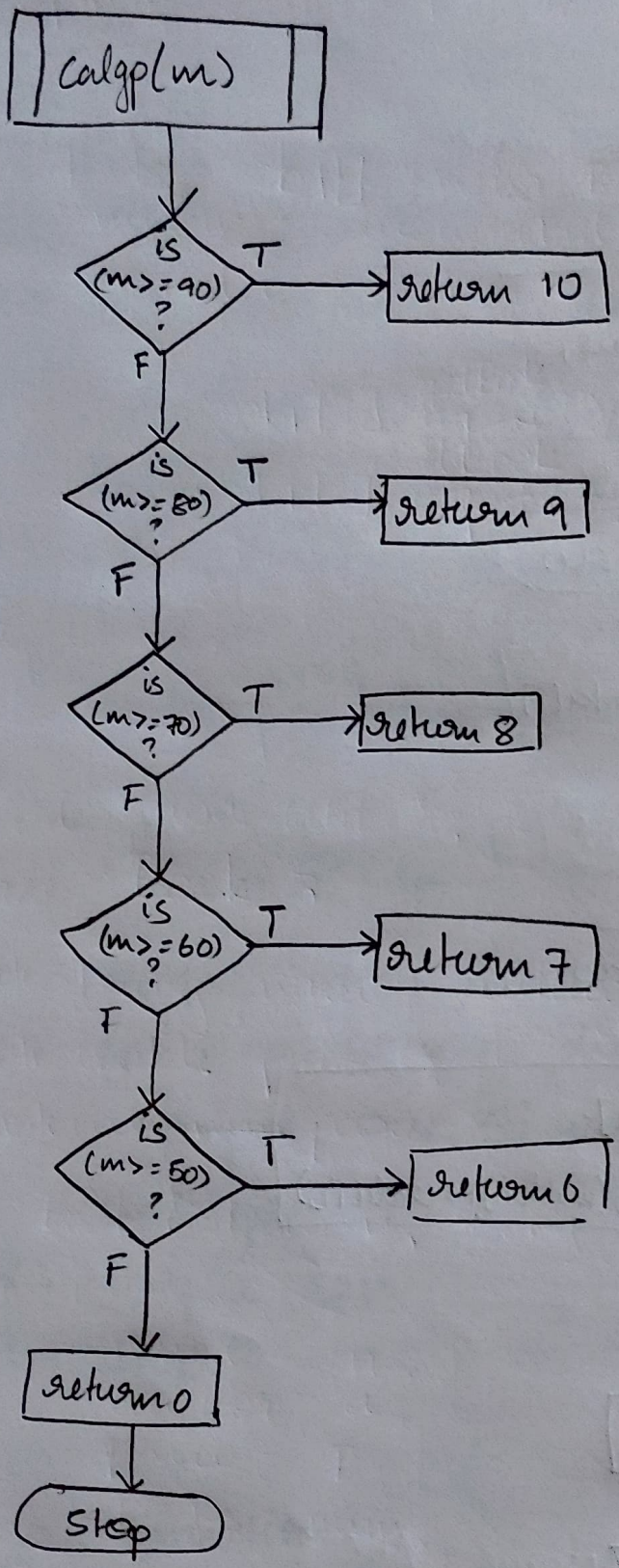
Step 1 : start

Step 2 : if marks >=90
return 10
else if marks >=80
return 9
else if marks >=70
return 8
else if marks >=60
return 7
else if marks >=50
return 6
else
return 0
endofif

Step 3 : stop

Flowchart :


Start
input usn, name, n
for i=0 to n-1, do
input credits[i], marks[i]
i
for i=0 to n-1, do
tc = tc+credits[i];
tgp = tgp+(call function calgp(marks[i]) * credit[i])
i
res = tgp/tc
output usn, name
for i=0 to n-1, do
output credit[i], marks[i]
i
output res
stop


calgp(m)
is (m>=90) ?
T
return 10
F
is (m>=80) ?
T
return 9
F
is (m>=70) ?
T
return 8
F
is (m>=60) ?
T
return 7
F
is (m>=50) ?
T
return 6
F
return 0
Stop

```
C:\Users\BMSCE\Desktop\014>javac Student.java

C:\Users\BMSCE\Desktop\014>java Student

Enter name: ABC

Enter USN: 123

Enter no. of subjects: 5
Enter marks and credits:
Enter marks for subject 1: 90
Enter credits for subject 1: 4
Enter marks for subject 2: 86
Enter credits for subject 2: 3
Enter marks for subject 3: 95
Enter credits for subject 3: 4
Enter marks for subject 4: 89
Enter credits for subject 4: 1
Enter marks for subject 5: 98
Enter credits for subject 5: 1
Name : ABC
USN : 123
Subject 1 :      Marks= 90      Credits= 4
Subject 2 :      Marks= 86      Credits= 3
Subject 3 :      Marks= 95      Credits= 4
Subject 4 :      Marks= 89      Credits= 1
Subject 5 :      Marks= 98      Credits= 1

SGPA : 9.692307692307692
Name: Aditi C
USN: 1BM22CS014
```

Program 3: Create a class Book which contains 4 members: name, author, price, num-pages Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Develop a Java program to create n book objects.

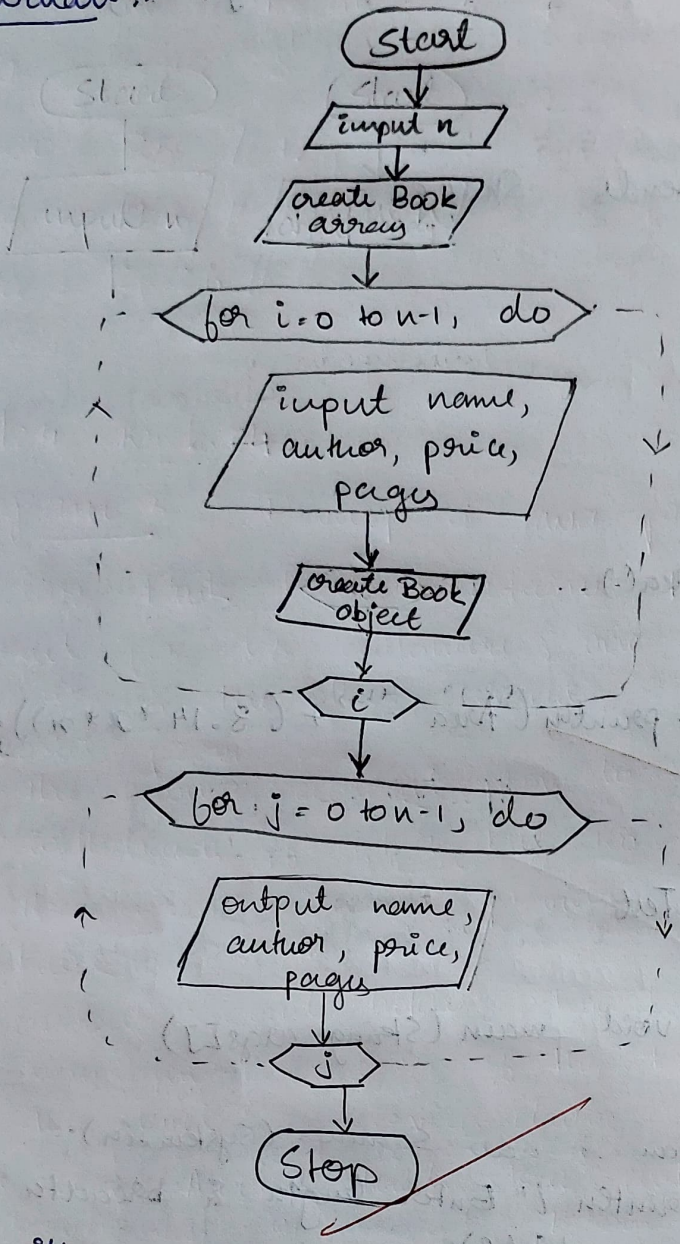
```
import java.util.*;
class Book
{
    string name;
    string author;
```

```
import java.util.*;
class Book {
private String name, author;
private double price;
private int num pages;
    Book (String n, String a, double p, int num)
    {
      name = n;
      author = a;
      price = p;
      pages = num;
    }
void setName (String neme)
  {
    this.name = neme;
  }
String getName()
{
  return name;
}
void setAuthor (String author)
{
  this.author = author;
}
String getAuthor ()
{
  return author;
}
```

```
void setPrice(double price)
{
    this.price = price;
}
double getPrice()
{
    return price;
}
void setPages(int pages)
{
    this.pages = pages;
}
int getPages()
{
    return pages;
}
String toString()
{
    return("Book Details: \n Name: " + name +
    "\n Author: " + author + "\n Price: " + price +
    "\n No. of pages: " + pages);
}
}

class BookTest {
public static void main(String args[]){
Scanner scan = new Scanner(System.in);
System.out.println("Enter no. of books: ");
int n = scan.nextInt();
Book ob[] = new Book[n];
for(int i=0; i<n; i++){
System.out.println("Enter "+(i+1)+" details: \n Name: ");
String na = scan.nextLine();
System.out.println("Author:"); String au = scan.nextLine();
System.out.println("Price:"); double p = scan.nextDouble();
System.out.println("No. of pages: "); int nn = scan.nextInt();
ob[i] = new Book(na, au, p, nn);
}
for(int i=0; i<n; i++){
System.out.println(ob[i].toString());
}}
```

Program 3 :-

Flowchart :-



Output:

Enter no. of books : 2
Enter 1 book details :
Name : The Great Gatsby
Author : F. Scott
Price : 700
No. of pages : 180

Enter 2 book details :
Name : Pride & Prejudice
Author : Jane Austen
Price : 1050
No. of pages : 400

Algorithm:

Step 1 : Start
Step 2 : input n
Step 3 : create Book array
Step 4 : for i = 0 to n-1, do
  input name, author, price, pages
  create Book object
  endoffor
Step 5 : for j=0 to n-1, do
  output name, author, price, pages
  endoffor
Step 6 : stop

```
C:\Users\BMSCE\Desktop\014>javac Book.java

C:\Users\BMSCE\Desktop\014>java Book
enter the number of books
2
Enter name
The Great Gatsby
Enter Author
F Scott
Enter Price
700
Enter Number of pages
180
Enter name
Pride and Prejudice
Enter Author
Jane Austen
Enter Price
1050
Enter Number of pages
400
THE BOOK LIBRARY
Name of book: The Great Gatsby
Name of author: F Scott
Price of book: 700
number of pages of book: 180
Name of book: Pride and Prejudice
Name of author: Jane Austen
Price of book: 1050
number of pages of book: 400
Name: Aditi C   USN:1BM22CS014
```

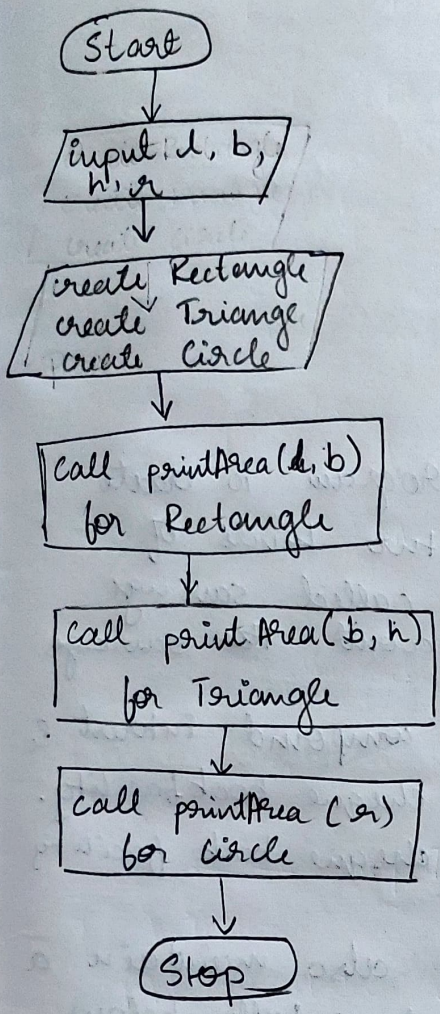
Program 4: WAP to create an abstract class named shape that contains two integers & an empty method named printArea(). Provide three classes named rectangle, triangle & circle such that each one of the class extends the class shape. Each one of the classes contains only the method printArea() that prints the area of given shape.

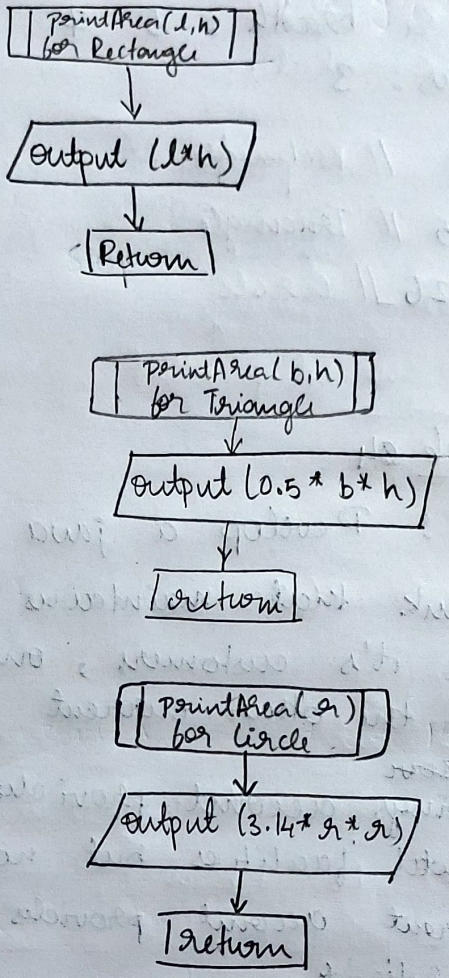
```
import java.util.*;
abstract class Shape
{
    int x;
    int y;
    abstract void printArea();
}
class Rectangle extends Shape
{
    void printArea(int l, int b)
    {
        x = l; y = b;
    }
    void printArea()
    {
        System.out.println("Area=" + (l*b));
    }
}
class Triangle extends Shape
{
    Triangle (int l, int h)
    {
        x = l;
        y = h;
    }
    void printArea()
```

```
    {
        System.out.println("Area = " + ((x*y)/2));
    }
}

class Circle extends Shape
{
    Circle()
    {
        r = r;
    }
    void printArea()
    {
        System.out.println("Area = " + (3.14 * r * r));
    }
}

class ~~Main~~ ShapeTest
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter length & breadth");
        int a = scan.nextInt();
        int b = scan.nextInt();
        ~~Rectangle~~ Shape A = new Rectangle(a, b);
        ~~Triangle~~ Shape B = new Triangle(a, b);
        System.out.println("Enter radius");
        int r = scan.nextInt();
        Shape ~~Circle~~ C = new Circle(r);
        A.printArea();
        B.printArea();
        C.printArea();
    }
}
```

Program 4 :-

Flowchart :-





Algorithm :-

Step 1 : Start
Step 2 : input l, b, h, r
Step 3 : call printArea(l,b) for Re
Step 3 : create Rectangle
Step 4 : create Triangle
Step 5 : create circle
Step 6 : call printArea(l,b) for Rectangle
Step 7 : Call printArea(b,h) for ~~Recta~~ Triangle
Step 8 : call printArea(r) for circle
Step 9 : Stop

printArea(l,b) for Rectangle
Step 1 : output (l*b)
Step 2 : return

printArea(b,h) for Triangle
Step 1 : output (0.5 * b * h)
Step 2 : return

printArea (r) for Circle
Step 1 : output (3.14 * r * r)
Step 2 : return

Output:-

Enter length & breadth : 5 7
Enter radius : 3

Area = 35 // Rectangle
Area = 17.5 // Triangle
Area = 28.26 // circle

```
C:\Users\BMSCE\Desktop\014>javac ShapeTest.java

C:\Users\BMSCE\Desktop\014>java ShapeTest
Enter length and breadth of rectangle:
5 7
Rectangle Area is 35
Enter length and height of triangle:
5 7
Triangle Area is 17.5
Enter radius:
3
Circle Area is 28.259999999999998
Name: Aditi C   USN:1BM22CS014
```

19/11/24 Week 04

Program 5 : Develop a java program to create a class Bank that maintains two kinds of account for it's customers, one called savings account & the other current account. The savings account prov

* The savings account provides compond interest & withdrawal facilities but no cheque book facility.
* The current account provides cheque book facility but no interest.
* Current account holders should also maintain a minimum balance & if the balance falls below this level, a service charge is imposed.

Create class Accnt that stores customer-name, account number & type of account. From this derive the class cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer & update balance
b) Display the balance
c) Compute & deposit interest
d) Permit withdrawal & update balance
e) Checks for the minimum balance, impose penalty if necessary & update balance.

```
import java.util.*;
class Bank & Account {
    String name;
    int acc;
    double bal;
    Account (String n, int a, double b)
    {
        name = n;
        acc = a;
        bal = b;
    }
    void deposit(double amt)
    {
        bal += amt;
        System.out.println("Deposit done. \n Current
        Balance : " + bal);
    }
    void printBal()
    {
        System.out.println("Current Balance = Rs " + bal);
    }
}
class curAcc extends Account{
    double minbal;
    boolean chq;
    curAcc(String name, int acc, double bal, double
        min, boolean chq, double c)
    {
        super(name, acc, bal);
        minbal = min;
        chq = ch;
        charge = c;
    }
    void cheque()
    {
        System.out.println("Cheque Book avialable = " + chq);
    }
```

```
void check ()
{
    if (bal < minbal)
    {
        System.out.println (" Balance below minimum!!
                            charge + " charge imposed ");
        bal -= charge;
        System.out.println(" charge deducted from balance ");
    }
    exit(0);
}

void withdraw (double amt)
{
  this.check ();
  if(amt < bal)
  { bal -= amt;
    System.out.println (" Withdrawn successful In Current
    Balance = Rs " + bal );
  }
  else
    System.out.println (" withdraw unsuccessful");
}

class SavAcc extends Account{
    double interestrate, t;
    boolean chq;
    SavAcc (String name, int acc, double bal, double i,
            double t, boolean ch)
    {
        super(name, acc, bal);
        interestrate = i; this.t = t;
        chq = ch;
    }
    void deposit (double amt)
    {
        super.deposit(amt);
        super.printBal();
    }
    void cint ()
    {
        double ci = bal * Math.pow((1+interestrate), t) - bal;
        bal += ci;
```

```
System.out.println(" Compound Interest = Rs " + ci +
"\n Current Balance = Rs " + bal);
}
void cheque()
{
System.out.println(" Cheque Book avialable = " + chq);
}
}

public class Bank {
public static void main(String args[]) {
CurAcc a[] = new CurAcc[10];
SavAcc s[] = new SavAcc[7];
Account a[] = new Account[];
Scanner scan = new Scanner(System.in);
System.out.println("Enter customer details :");
System.out.println("Enter name : ");
String name = scan.nextLine();
System.out.println("Enter accoundnumber = ");
int ace = scan.nextInt()
System.out.println("Enter balance = ");
double bal = scan.nextDouble();
Account a = new Account(name, ace, bal);
System.out.println("Enter type of account\n1. Savings
\n2. Current ");
int ch = scan.nextInt();
Switch(ch)
{
case 1: {System.out.println("Enter interestrate = ");
double i = scan.nextDouble();
System.out.println("Enter term in years = ");
double t = scan.nextDouble();
SavAcc s = new SavAcc(name, ace, bal,
System.out.println("Enter withdraw amt:");  i/100, t, false);
double amt = scan.nextDouble(); s.cint();
s.withdraw(amt);
s.cheque();
} break;
```

```
case 2: { System.out.println("Enter min balance = ");
          double m = scan.nextDouble();
          System.out.println("Enter service charge = ");
          double c1 = scan.nextDouble();
          Current c = new Current(name, acc, bal, m,
              true, c1);
          double amt = scan.nextDouble();
          c.cheque();
          c.withdraw(amt);
        } break;
default: System.out.println("Invalid choice");
    }
  }
}
```

Output:

```
Enter customer details:
Enter name : ABC
Enter account : 123
Enter balance : 10000
Enter type of account :
1. Saving
2. Current
2
check available = false
Enter interest rate = 8
Enter term in years = 1
Compound interest = Rs. 800.0
Current Balance = 10800.0
Enter withdraw amt = 500
Withdrew Successful
Current Balance = 10300.0
```

output (ii):

Enter customer details:
Enter name: BCD
Enter account: 245
Enter balance: 50000
Enter type of account:
1. Savings
2. Current
2.
Cheque book available = true
Enter min balance = 500
Enter service charge = 50
Enter withdraw amount = 1500
cheque book available = true
withdraw successful
Current Balance = 48500

19/01/24

Algorithm

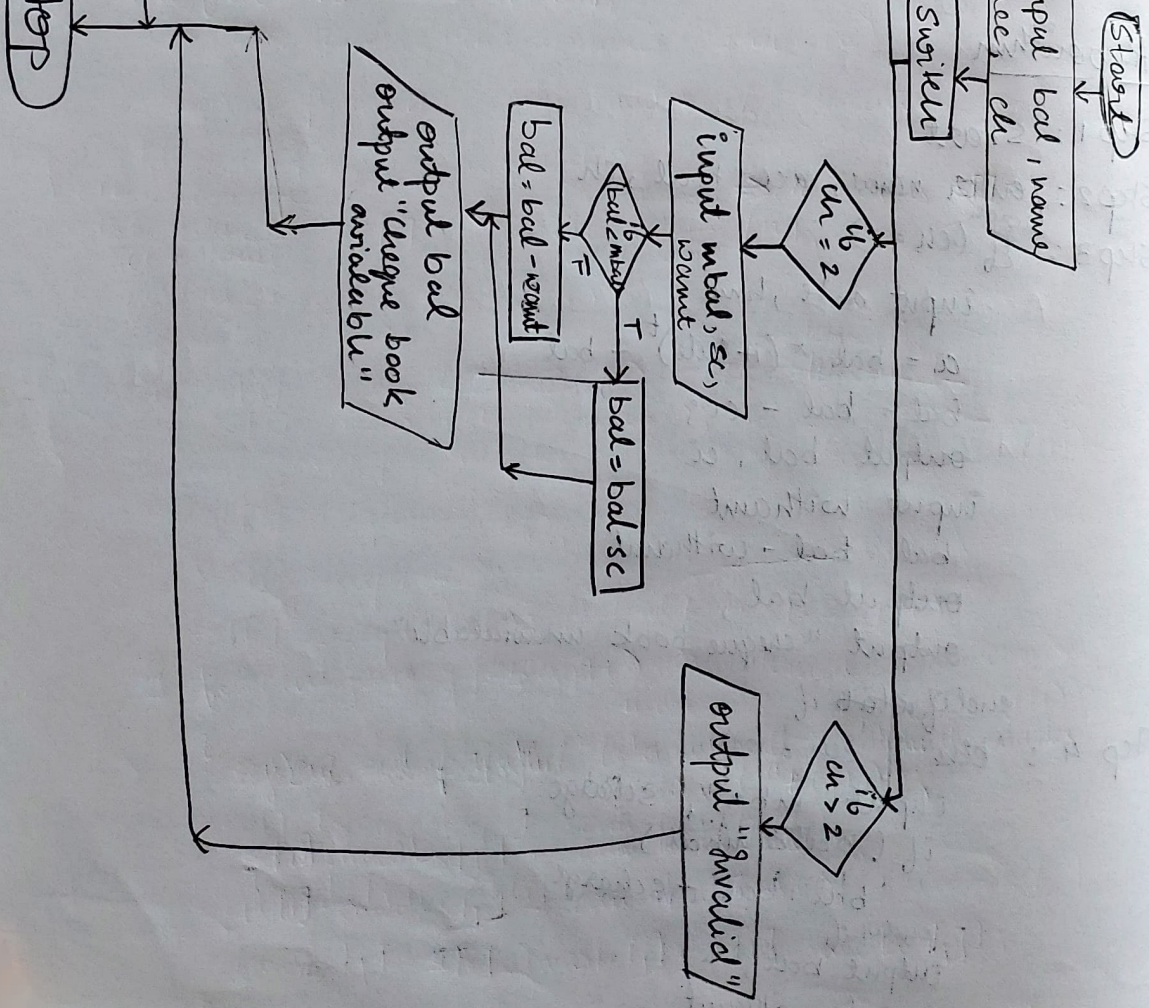
Step 1: Start
Step 2: enter name, acc, bal, ch
Step 3: switch (ch = 1)
input rate, time
ci = bal * (1+rate)^t - bal
bal = bal - ci;
output bal, ci
input withamt
bal = bal - withamt
output bal;
output "cheque book unavailable"
endif if
Step 4: else if (ch = 2)
input mbal, scharge
if (bal < mbal)
bal = bal - scharge
endif
output bal;
input withamt
bal = bal - withamt

output bal
output "cheque book avidable"
endif;

Step 5: else
output invalid

Step 6: stop

Flowchart:-



```
C:\Users\BMSCE\Desktop\014>javac Bank.java

C:\Users\BMSCE\Desktop\014>java Bank
Enter customer details
Enter name:
ABC
Enter account number:
123
Enter balance:
15000
Enter type of account
1. Savings
2. Current
1
Calculating compound interest
Enter interest rate:
15
Enter time in terms of years:
3
Compound interest= Rs 7813.124999999996
Current Balance= Rs 22813.124999999996
Do you wish to deposit or withdraw?
Enter: 1-withdraw       2-deposit
2
Enter withdraw amt:
500
Withdrawal success
Current Balance: 22313.124999999996
Current Balance = Rs 22313.124999999996
Cheque book available: false
Name: Aditi C   USN:1BM22CS014
```

```
C:\Users\BMSCE\Desktop\014>javac Bank.java

C:\Users\BMSCE\Desktop\014>java Bank
Enter customer details
Enter name:
ABC
Enter account number:
123
Enter balance:
50000
Enter type of account
1. Savings
2. Current
2
Checking for minimum balance
Do you wish to withdraw?(1/0)
1
Enter withdraw amt:
10000
Withdrawal success
Current Balance: 40000.0
Current Balance = Rs 50000.0
Cheque book available: true
Name: Aditi C   USN:1BM22CS014
```

2/2/24

Week 5:

Program 6: Create a package CIE which has 2 classes - Student & Internals. The class Student has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all 5 courses.

```
Package CIE;
public class Student {
    public String usn, name;
    public int sem;
    public void acceptdetails()
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter name, usn & semester :");
        name = scan.nextLine();
        usn = scan.nextLine();
        sem = scan.nextInt();
    }
}
```

```
Package CIE;
public class Internals
{
public int imarks[] = new int[5];
    public void acceptint()
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter internals marks :\n");
        for(int i=0; i<5; i++)
            imarks[i] = scan.nextInt();
    }
```

```
package SEE;
import CIE.Student;
public class External extends Student{
   public int smarks[] = new int[5];

   public void acceptext()
   {
      Scanner scan = new Scanner(System.in);
      System.out.println("Enter external marks:\n");
      for(int i=0; i<5; i++)
         smarks[i]=scan.nextInt();
   }
}
```

```
import CIE.*;
import SEE.*;
import java.util.*;
public class FinalMarks{
   public static void main(String args[])
   {  int fmarks[] = new int[5];
      Scanner scan = new Scanner(System.in);
      System.out.println("Enter number of students : ");
      int n = scan.nextInt();
      SEE.External st[] = new SEE.External[n];
      CIE.Internals s[] = new CIE.Internals[n];
      for(int i=0; i<n; i++)
      {
         st[i] = new SEE.External();
         s[i] = new CIE.Internals();
         System.out.println("Enter details of student "+(i+1));
         st[i].acceptdetails();
         st[i].acceptint();
         st[i].acceptext();
      }
      for(int j=0; j<5; j++)
      {
         System.out.println("Enter internal & external marks
                             of subject "+(j+1));
         s[i].imarks[j] = scan.nextInt();
         st[i].smarks[j] = scan.nextInt();
         fmarks[j] = s[i].imarks[j]+st[i].smarks[j];
      }
```

```
System.out.println("Final marks of " + st[i].name);
for(int k=0; k<5; k++)
{
    System.out.println("Course " +(k+1)+ " = " + fmarks[k]);
}
}
}
```

output:-

```
Enter number of students: 1
Enter details of student 1:
Enter USN, name & sem:
123
ABC
3
Enter internal & external marks of subject 1
48
49
Enter internal & external marks of subject 2
48
50
Enter internal & external marks of subject 3
48
48
Enter internal & external marks of subject 4
49
49
Enter internal & external marks of subject 5.
50
50
Final marks of ABC:
Course 1 = 97
Course 2 = 98
Course 3 = 96
Course 4 = 98
Course 5 = 100
```

2/2/24

Algorithm :-

Step 1: Create package CIE, SEE
Step 2: Create class Student within CIE
Step 3: Create class Internals within CIE
Step 4: Create class External within SEE which extends CIE.Student
Step 5: Create method accept() which reads usn, name and sem.
Step 6: for i=0 to n-1 do
    call accept() for student i
    for j=0 to 4 do
        input imarks[j] for subject j of student i
        input smarks[j] for subject j of student i
        fmarks = imarks[j] + smarks[j]
    endfor
    output, "Final marks of student ", i
    for k=0 to 4 do
        output, "Course ", (k+1), " = ", fmarks[k]
    endfor
endfor
Step 7: Stop

```
C:\Users\BMSCE\Desktop\014\Package1>javac -d .  Student.java

C:\Users\BMSCE\Desktop\014\Package1>javac -d .  Internals.java

C:\Users\BMSCE\Desktop\014\Package1>javac -d .  External.java
```

```
C:\Users\BMSCE\Desktop\014\Package1>javac FinalMarks.java

C:\Users\BMSCE\Desktop\014\Package1>java FinalMarks
Enter number of students:
2
Enter details of student: 1
Enter USN, name and sem:
123
ABC
3
Enter internal and external marks of subject 1
48
49
Enter internal and external marks of subject 2
48
50
Enter internal and external marks of subject 3
48
48
Enter internal and external marks of subject 4
49
49
Enter internal and external marks of subject 5
50
50
Final marks of ABC
Course 1= 97
Course 2= 98
Course 3= 96
Course 4= 98
Course 5= 100
Enter details of student: 2
Enter USN, name and sem:
124
DEF
3
Enter internal and external marks of subject 1
48
48
Enter internal and external marks of subject 2
49
50
Enter internal and external marks of subject 3
50
50
Enter internal and external marks of subject 4
59
50
Enter internal and external marks of subject 5
48
49
Final marks of DEF
Course 1= 96
Course 2= 99
Course 3= 100
Course 4= 109
Course 5= 97
Name: Aditi. C   USN:1BM22CS014
```

10/2/24

WEEK

Program 7 : Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" & derived class "son" which extends the base class. In Father class, implement a constructor which takes the age & throws the exception WrongAge() when input age <0. In Son class, implement a constructor that cases both father & Son's age & throws an exception if Son's age is >= father's age.

```
import java.util.*;

class WrongAge extends Exception
{   String msg;
    WrongAge (String msg)
    {   this.msg = msg;
        System.out.println(msg);
    }
}
Class Father throws WrongAge
{ int age;
   Father(int age)
    {
      if (age<0)
        throw new WrongAge("Age can't be negative");
      this.age =age;
    }
   int getAge()
   {
     return age;
   }
}

Class Son extends Father throws WrongAge
{ int sage;
   Son (int fage, int sage)
    {
      super(fage);
```

```
    if (fage <= sage)
    {
      throw new WrongAge("father's age can't be less than
                                son's age");
    }

    this.sage = sage;
  }

  int getSAge()
  {
    return sage;
  }
}

class AgeTest {
{
  public static void main(String args[])
  {
    Scanner sc = new Scanner(System.in);
    int f, s;
    System.out.println("Enter age of father & son");
    f = sc.nextInt();
    s = sc.nextInt();
    try
    {
      Father obj1 = new Father(f);
      Son obj2 = new Son(f, s);
    }
    catch (WrongAge e)
    {
      System.out.println("Exception & " );
    }

  }
}
```

output :

i) Enter age of father & son : 50 18
// no exception

(ii) Enter age of father and son : 18 50
Father age cannot be less than Son's age
Exception thrown

(iii) Enter age of father and son : -12 20
Age cannot be negative
Exception

Algorithm:-

Step 1: start

Step 2: create user-defined class Exception WrongAge which prints the exception thrown by Father & Son classes

Step 3: create class Father which takes inputs fage and checks if age entered is negative & throws exception

Step 4: create class Son which extends Father class, this takes inputs fage & sage and checks if fage is <= sage and throws expception.

Step 5: In main class we create objects for Father & Son and passes user inputs for ages (A try-catch block looks for any exception thrown).

Step 6: stop.

* Father & Son classes have a throws exception in their declaration (they might throw user defined Exception : WrongAge)

```
Enter ages of father and son:
50
18
Name: Aditi C
USN: 1BM22CS014

C:\Users\BMSCE\Desktop\014>javac Demo.java

C:\Users\BMSCE\Desktop\014>java Demo
Enter ages of father and son:
18
50
Father age cannot be less than Son's age
Exception
Name: Aditi C
USN: 1BM22CS014

C:\Users\BMSCE\Desktop\014>javac Demo.java

C:\Users\BMSCE\Desktop\014>java Demo
Enter ages of father and son:
-12
20
Age can't be negative
Exception
Name: Aditi C
USN: 1BM22CS014
```

Program 8: WAP which creates 2 threads, one thread displaying "BMS college of Engineering" once every ten seconds & another displaying "CSE" once every two seconds.

```
import java.util.*;
class NewThread implements Runnable
{
  String name;
  Thread t;
  NewThread(String n)
  {
    name = n;
    t = new Thread(this, name);
    t.start();
  }
public void run()
{ for(int i=0; i<5; i++)
  if (name=="one")
  { System.out.println("BMS College of Engineering");
    try
    {
      Thread.sleep(10000);
    }
    catch (InterruptedException e)
    {
      System.out.println("Interrupted");
    }
  }
else if (name=="two")
  { System.out.println("CSE");
    try
    {
      Thread.sleep(2000);
    }
    catch (InterruptedException e)
    { System.out.println("Interrupted"); }
```

~~System.out.println ("CSE");~~

}
}
}

```
class Demo {
public static void main (String args[])
{
    NewThread obj1 = new NewThread ("One");
    NewThread obj2 = new NewThread ("two");
    try
    {
      obj1.t.join();
      obj2.t.join();
    }
    catch (InterruptedException e)
      { System.out.println ("Main thread, interrupted"); }
  }
}
```

Output:

```
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
BMS college of Engineering
BMS college of Engineering
BMS college of Engineering
```

Algorithm:-

Algorithm :-

Step 1 : Start

Step 2 : create a class NewThread that implements the Runnable interface, which takes name of the thread and starts the thread (using start())

Step 3 : the start() method initiates the run() method

Step 4 : in the run method :

```
for i = 0 to 4 do
    if (thread name = "one")
        output "BMS college of Engineering"
        thread sleeps for 10 seconds
    else if (thread name = "two")
        output "CSE"
        thread sleeps for 2 seconds
    endif
endfor
```

Step 5 : Stop

* A try and catch block in main function catches an interrupted exception

16.02.24

```
C:\Users\BMSCE\Desktop\014>javac Test.java

C:\Users\BMSCE\Desktop\014>java Test
BMS Ccollege of engineering
CSE
CSE
CSE
CSE
CSE
BMS Ccollege of engineering
BMS Ccollege of engineering
BMS Ccollege of engineering
BMS Ccollege of engineering
Name: Aditi C
USN: 1BM22CS014
```

23/2/24

Program 9: WAP that creates a user interface to perform integer divisions. The user enters 2 numbers in the text fields, Num1 & Num2. The division of Num1 & Num2 is displayed in the Result field when the divide button is clicked. If Num1 & Num2 were not an integers, the program would throw a NumberFormatException. If Num2 were a zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

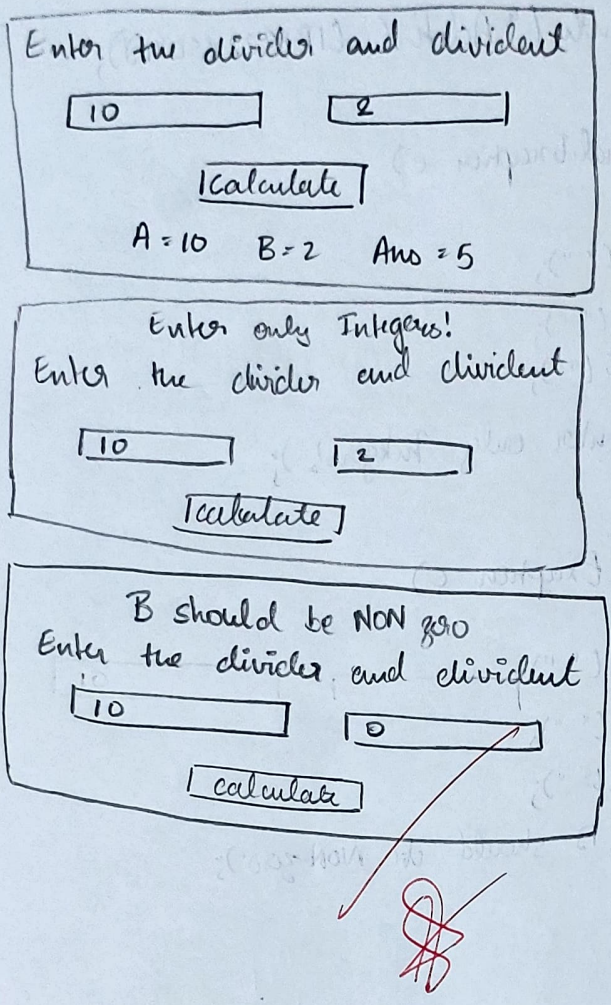
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo
{
  SwingDemo()
  {
    JFrame jfrm = new JFrame("Divider App");
    // A class in java which impact the JFrame
    jfrm.setSize(265, 150);   // width, height
    jfrm.setLayout(new FlowLayout());
    // arranges components in left to right flow.
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //sets the close operation
    JLabel jlab = new JLabel("Enter the divider and dividend : ");
    // component that displays a short text string, an img or both
    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);
    // allows editing of single line of text
    // '8' specifies the width of the text field in terms of column
    JButton button = new JButton("Calculate");
    // a push button that performs a specific task
    JLabel err = new JLabel(); // initially empty but later used to display error
    JLabel alab = new JLabel(); // to display component or
```

```
JLabel blab = new JLabel(); // to display b
JLabel anslab = new JLabel(); // to display ans

// add in order in the frame
jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);


ActionListener I = new ActionListener()
{
    public void actionPerformed(ActionEvent evt)
    {
        System.out.println("Action event from text field");
    }
};

/* ActionListener : an interface in Swing framework that
listens for & handles action events.
actionPerformed: part of ActionListener interface which handles
action events like button clicks or text field entries. */

ajtf.addActionListener(I);
bjtf.addActionListener(I);

button.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent evt)
    {
        if(err.getText() != NULL)
            err.setText("");
        try
        {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;
            alab.setText("\nA = " +a);
```

```
        blab.setText("\nB = " + b);
        anslab.setText("\nAns = " + ans);
        System.out.println("Aditi.C (1BM22CS041)");
    }
    catch (NumberFormatException e)
    {
        alab.setText(" ");
        blab.setText(" ");
        anslab.setText(" ");
        err.setText("Enter only Integers!.");
    }
    catch (ArithmeticException e)
    {
        alab.setText(" ");
        blab.setText(" ");
        anslab.setText(" ");
        err.setText("B should be NON ZERO");
    }
  }
});
//display frame
jfm.setVisible(true);
}
public static void main(String args[])
{
// creating from an event dispatching thread
SwingUtilities.invokeLater(new Runnable()
{
    public void run()
    {
        new SwingDemo();
    }
});
}
}
```

23/12

Output:-



JFrame functions:-

JFrame : A class in java which impacts JFrame.

JLabel : component that displays a short text, or an image or both

setSize : set size of frame (int width, int height)

setDefaultCloseOperation: sets the close operation

JTextField(8): allows editing of single line of text
'8' is the width of text in terms of column

JButton : a push button that performs a specified task

add( ) : add the components to the frame

ActionListener: an interface in Swing framework that listens for and handles action event

actionPerformed: part of ActionListener interface which handles action events like button clicks or text field entries

setText(): sets the text content of the Swing component to the specified string

getText(): retrieves the text content from the Swing component

setVisible(): sets the visibility of the JFrame

SwingUtilities.invokeLater: used to execute code that updates Swing components

Divider app

Enter the divider and divident:

10

2

Calculate

A= 10 B= 2

Ans= 5

Aditi C [1BM22CS014]

Divider app

B should be NON zero!

Enter the divider and divident:

10

0

Calculate

Aditi C [1BM22CS014]

Divider app

Enter only Integers!

Enter the divider and divident:

10

a

Calculate

Aditi C [1BM22CS014]